Constraint Programming for Itemset Mining

Luc De Raedt - Tias Guns – Siegfried Nijssen

Itemset Mining

Input

Items: x, y, z
Transactions:

	x	y	z
{x,y}	1	1	0
{x,y,z} or	1	1	1
{x,y}	1	1	0
{y,z}	0	1	1

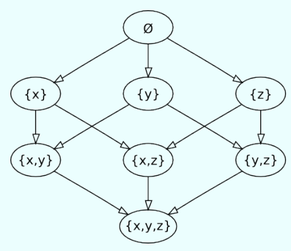
Output

Frequent (min. 3) Itemsets:

{a} freq=3
{a,d} freq=3
{d} freq=4

Search space

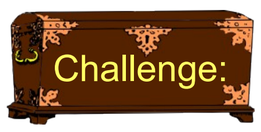
lattice wrt. subset relation:



level-wise enumeration:

- anti-monotone constraints
- others...

Constraint-based Itemset Mining



push constraints into the mining process

- Itemset constraints (minimum size, average cost,...)
- Condensed representations (closed, maximal itemsets,...)
- Alternatives for support (emerging, correlated patterns,...)

...many, many, many constraints and algorithms proposed...

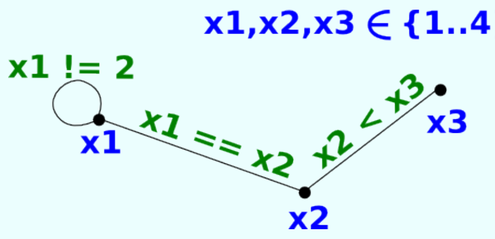
BUT no **generic framework**

Constraint Programming

CP is a general approach to combinatoral problem solving

Input

a model: variables & constraints

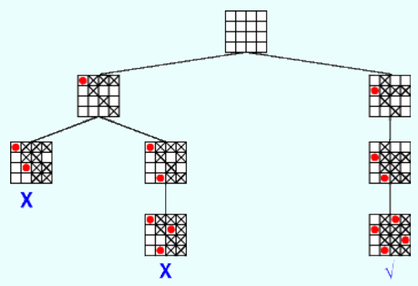


Output

all valid assignments:
{x1,x2,x3} = {1,1,2},{1,1,3}, {3,3,4}, ...

Search space

example 4-queens:



constraint-search:

- *fix* a variable
- *propagate* constraints
  - if failure: *backtrack*
  - else: *contraint-search*

Propagation

- a variable has a domain
- a constraint has a propagator

a propagator can **reduce** the domain of its variables.

Power of CP:

combined effect of propagators

modelling language

declarative and very expressive

FIM_CP: Constraint Programming as Generic Framework for IM

Algorithm 1 Fim_cp's frequent itemset mining model, in Essence'

```
1: given NrT, NrI : int
2: given TDB : matrix indexed by [int(1..NrT),int(1..NrI)] of bool
3: given Freq : int
4: find Items : matrix indexed by [int(1..NrI)] of bool
5: find Trans : matrix indexed by [int(1..NrT)] of bool
6: such that
7: $ encode TDB: every Trans its complement has no supported Items
8: forall t: int(1..NrT).
9:   Trans[t] <=> ((sum i: int(1..NrI). !TDB[t,i]*Items[i]) <= 0),
10: $ frequency: every Item is supported by sufficently many Trans
11: forall i: int(1..NrI).
12:   Items[i] => ((sum t: int(1..NrT). TDB[t,i]*Trans[t]) >= Freq)
```

Essence' is a known solver-independant constraint modelling language.

(8-9) : $\forall t \in \mathcal{T} : T_t = 1 \quad \leftrightarrow \quad \sum_{i \in \mathcal{I}} I_i(1 - \mathcal{D}_{ti}) = 0.$

Propagation:
if one item=1 in transaction's complement → trans=0
if all items=0 in transaction's complement → trans=1

(10-11) : $\forall i \in \mathcal{I} : I_i = 1 \quad \rightarrow \quad \sum_{t \in \mathcal{T}} T_t \mathcal{D}_{ti} \geq \theta.$

Propagation:
if too many of item's trans=0 → item=0

Many more constraints possible!
(average cost, delta-closed, ...)

Arbitrary combinations of them
(also diff. interpretations)

Conclusion

- Wide range of constraints out-of-the-box
- The modelling power of CP for IM
  eg. combining constraints
- Also good on dense datasets
  - Not so good on very large datasets

Future Work

- Even more constraints
- Optimise CP algorithms for IM
- Other types of patterns
  eg. correlated itemsets or trees, graphs, ...

KATHOLIEKE UNIVERSITEIT LEUVEN
DTAI
04.034
<Luc . DeRaedt @cs.kuleuven.be>
<Tias . Guns @cs.kuleuven.be>
<Siegfried . Nijssen @cs.kuleuven.be>

CP4IM http://www.cs.kuleuven.ac.be/~dtai/CP4IM/ CP4IM